Input buffer

The **Input buffer** is also commonly known as the **input area** or **input block**.

1.When referring to computer memory, the input buffer is a location that holds all incoming information before it continues to the CPU for processing.

2.Input buffer can be also used to describe other hardware or software buffers used to store information before it is processed.

The sentinel is a special character that cannot be part of the source program, and a natural choice is the character eof. ... Any eof that appears other than at the end of a **buffer** means that the **input** is at an end

The usage of sentinel reduces the two tests to one by extending each buffer half to hold a **sentinel** character at the end. The **sentinel** is a special character that cannot be part of the source program.

The End-of-File value is a special value used by many standard functions to represent an invalid **character**; Its value shall not compare equal to any of the values representable with char_type .

Why do we use buffer pairs?

Buffer Pairs: Because of the amount of time taken to process characters and the large number of characters that **must** be processed during the compilation of a large source program, specialized buffering techniques **have** been developed to reduce the amount of overhead required to process a single input character.

Two pointers to the input are maintained:

1. **Pointer lexeme begin**: which is marks the beginning of the current lexeme. It marks the beginning of the current lexeme, whose extent we are attempting to determine.
2. **Pointer forward:** which is scans ahead until a pattern match is found.

Problems of buffer pair:

1. Advancing forward requires that we first test weather we have reached the end of one of the buffers.
2. If so we must reload the other buffer from the input.
3. And move forward to the beginning of the newly loaded buffer.

What is transition diagram in Compiler?

Transition diagram is a special kind of flowchart for language analysis.
In **transition diagram** the boxes of flowchart are drawn as circle and called as states. States are connected by arrows called as edges. The label or weight on edge indicates the input character that can appear after that state.

a **reserved word** (also known as a **reserved** identifier) is a **word** that cannot be used as an identifier, such as the name of a variable, function, or label – it is "**reserved** from use". This is a syntactic definition, and a **reserved word** may have no meaning. For example print.

Recognition of tokens:

Is the pattern using regular expression.

Regular expression is an important notation for specifying patterns. ... Each pattern matches a set of strings, so **regular expressions** serve as names for a set of strings. Programming language tokens can be described by **regular** languages.

Example:

Digit--→[0 to 09]
Digits→digit+
Number-→digit(.digits)?(E[+-]?digits)?
Letter →[A-Za-Z]
Id→letter(letter digit)*
If→if
Then →then
Else →else
Relope → </>/<=/>=/=/<>

Identifiers are symbols used to uniquely identify a program element in the code. They are also used to refer to types, constants, macros and parameters.

2 ways that we can handle reserved words :

- **Installed the reserved words in the symbol table**
- **Create separate transition diagrams for each keyword.**

whitespace is any **character** or series of **characters** that represent horizontal or vertical space in typography. When rendered, a **whitespace character** does not correspond to a visible mark, but typically does occupy an area on a page.

Variable:

Variable is the name of memory location where it is loaded a value.

Function of next character: It obtains the next character from the input and assigns it to local variable.

Failed function: if the next input character is not one that can begin a comparison operator is called fail function.

LEXEMES	TOKEN NAME	ATTRIBUTE VALUE
Any *ws*	–	–
`if`	**if**	–
`then`	**then**	–
`else`	**else**	–
Any *id*	**id**	Pointer to table entry
Any *number*	**number**	Pointer to table entry
`<`	**relop**	LT
`<=`	**relop**	LE
`=`	**relop**	EQ
`<>`	**relop**	NE
`>`	**relop**	GT
`>=`	**relop**	GE

Tokens, their patterns, and attribute values

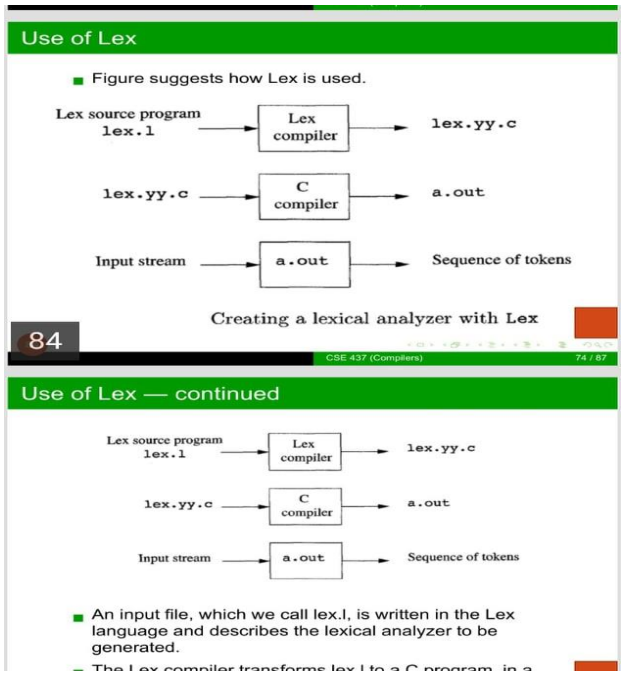
- Our goal for the lexical analyzer is summarized in figure.

37

Lex is a computer program that generates lexical analyzers ("scanners" or "lexers"). **Lex** is commonly used with the yacc parser generator

FLEX (fast lexical analyzer generator) is a tool/computer program for generating lexical analyzers (scanners or lexers) written by Vern Paxson

in C around 1987.



Lex compiler: The notation for the lex tool is referred to as the lex language and the tool itself is the lex compiler. a tool/computer program for generating lexical analyzers (scanners or lexers) written by Vern Paxson in C around 1987

Lex .yy.c:

The lex compiler transforms the input patterns into a transition diagram and generates code in a file called lex.yy.c.

The yylval global variable is used to pass the semantic value associated with a token from the lexer to the parser.

a. … **out"** remains the default output file name for executables created by certain **compilers** and linkers when no output name is specified, even though the created files actually are not in the a. **out** format.

Structure of a Lex Specification?

A **lex program** consists of three parts: the definition section, the rules section, and the user subroutines. The parts are separated by lines consisting of two percent signs. The first two parts are required, although a part may be empty.

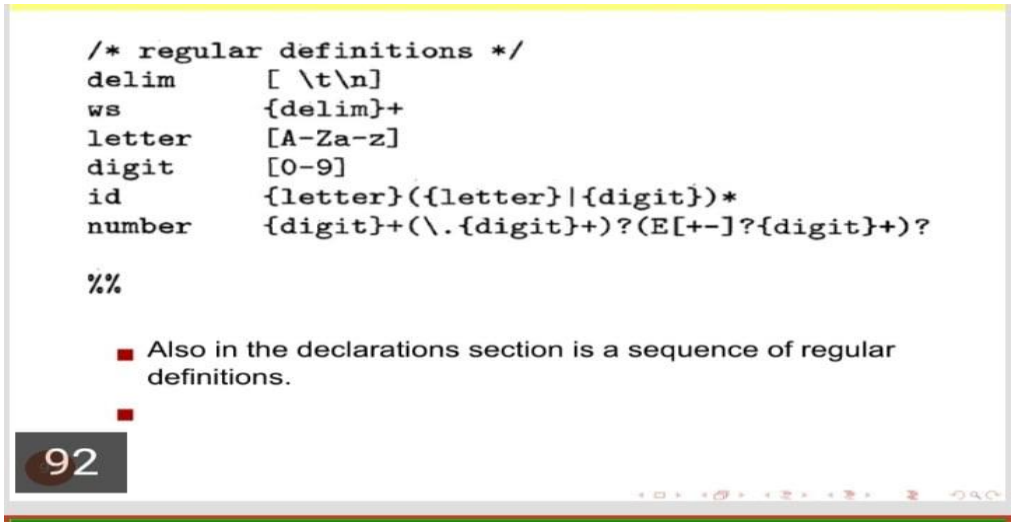
The three form of the lex program:

Declaration %%, translation rules %%, auxiliary function.

Explain all the component of lex program:

A **lex program** consists of three **parts**: the definition section, the rules section, and the user subroutines. The **parts** are separated by lines consisting of two percent signs. The first two **parts** are required, although a part may be empty. The third part and the preceding %% line may be omitted.

Regular definition:



Formula of translation rule:

The translation rule each have the form pattern(action).

Each pattern is a regular expression which may use the regular definition of the declaration section.

The actions are fragments of code, typically written in C.

Auxiliary functions **are** not a rigorously defined kind of **function**, rather **they are functions** which **are** either explicitly constructed or at least shown to exist and which provide a contradiction to some assumed hypothesis, or otherwise prove the result in question.

Yytext

holds the text matched by the current token. So **yytext**[0] holds the first character of the text matched by the current token. Sometimes **you** have a rule which **can** match different texts so **you** need to get the real text matched like for variable names or **you** have a rule to match all arithmetic operations.

yyleng is the length of the matched string.

Symbol

A symbol is a mark, sign, or word that indicates, signifies, or is understood as representing an idea, object, or relationship.

Alphabet

a set of letters or symbols in a fixed order used to represent the basic set of speech sounds of a language, especially the set of letters from A to Z.

Language

A language is a structured system of communication. Language, in a broader sense, is the method of communication that involves the use of – particularly human – languages.

String

a string is traditionally a sequence of characters, either as a literal constant or as some kind of variable.

Length of string

The **length** of a **string** is the number of symbols (counting duplicates) in the **string**. **Example**: The **length** of allan, written |allan|, is 5.

a **substring** is a contiguous sequence of characters within a **string**.

Terms for Parts of the string:

Prefix of string

Suffix of string, substring of string, proper prefix suffix and substring of a string, subsequence of string.

A prefix

is an affix which is placed before the stem of a word. Adding it to the beginning of one word changes it into another word. For **example**, when the **prefix** un- is added to the word happy, it creates the word unhappy. .

Suffixes

are a letter or group of letters added to the ending of words to change their meaning or function. These useful, shapeshifting tools can be as small as -s, and -ed, or can be larger additions such as -ation, and -ious.

What is proper prefix?

A **proper prefix** of a string is a **prefix** that is not equal to the string itself. By definition, $\pi[0]=0$.

A **proper suffix**

of a string is not equal to the string itself. A more restricted interpretation is that it is also not empty ...

a trie

, also called digital tree or prefix tree, is a kind of search tree—an ordered tree data structure used to store a dynamic set or associative array where the keys are usually strings.

AMP algorithm

The approximate message passing (**AMP**) **algorithm**, designed for compressed sensing, has attracted researchers to counter this problem due to its reduced complexity with a large system limit. ... These estimation functions are obtained using the log-sum approximation, then taking the exponent of the result.

Aho-corasick algorithm

the Aho–Corasick algorithm is a string-searching algorithm invented by Alfred V. Aho and Margaret J. Corasick. It is a kind of dictionary-matching algorithm that locates elements of a finite set of strings within an input text. It matches all strings simultaneously.

Operation on language:

On languages we can define the usual set **operations** that is union, intersection and complement of languages.

The UNION operator is used to combine the result-set of two or more SELECT statements. Each SELECT statement within **UNION** must have the same number of columns. The columns must also have similar data types. The columns in each SELECT statement must also be in the same order.

Concatenation

is the operation of joining character strings end-to-end. For example, the concatenation of "snow" and "ball" is "snowball".

A **closure** is the combination of a function bundled together (enclosed) with references to its surrounding state (the lexical environment). In other words, a **closure** gives you access to an outer function's scope from an inner function.

The notation Ω(n) is the formal way to express the lower bound of an **algorithm's** running time. It measures the best case time complexity or the best amount of time an **algorithm** can possibly take to complete. For example, for a function f(n)

Role of the parser : ... The **parser** obtains a string of tokens from the lexical analyzer and verifies that the string can be the grammar for the source language. It detects and reports any syntax errors and produces a **parse** tree from which intermediate code can be generated.

Expect of the parser:

To report any syntax error, to recover from commonly occurring errors to continue processing the remainder of the program.

Three general types of parser:

Universal,
top down and
bottom up.

Another name of universal parser:

Cocke younger kasami algorithm and earlys algorithm.

Disadvantage of **LR parsers** is that their tables can be very large.

What are the problems of top down parsing?

The following are the problems associated with top down parsing:

- Backtracking.
- Left recursion.
- Left factoring.
- Ambiguity.

Recursion

is a method of solving a problem where the solution depends on solutions to smaller instances of the same problem.

2 type of recursion:

Left recursion: left recursion is a special case of recursion where a string is recognized as part of a language by the fact that it decomposes into a string from that same language and a suffix.

Right recursion

A **recursive** grammar is said to be **right recursive** if the rightmost variable of RHS is same as variable of LHS.

Non recursive grammar:

a **grammar** is informally called a **recursive grammar** if it contains production rules that are **recursive**, meaning that expanding a **non**-terminal according to these rules can eventually lead to a string that includes the same **non**-terminal again. Otherwise it is called a **non-recursive grammar**.

Two types of derivation: 1.left most 2.right most

What is the formula of more than one recursion?

A→B1A'|B2A'|B3A'……BnA'